



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

LABORATORIO DE DISEÑO DIGITAL

Reporte de práctica 10

Divisores de tiempo

Alumno(s):

Francisco Pablo RODRIGO

Profesor:

M.I. Guevara Rodríguez MA. DEL

SOCORRO

Grupo: 6

Calificación total _____

Previo _____

Desarrollo _____

Conclusiones _____

6 de mayo de 2019

1. Objetivos

1.1. General

El alumno diseñará circuitos secuenciales.

1.2. Particular

Analizar, diseñar, simular e implementar divisores de tiempo.

2. Introducción

Los circuitos digitales, a no ser que sean asíncronos, van comandados por un reloj cuya frecuencia puede variar según el tipo de sistema digital del que se trate. Desde microprocesador 6502 que funcionaba con un reloj de 1Mhz hasta los actuales, que funcionan en el orden de los gigahercios, no han pasado ni cuatro décadas. En un sistema digital complejo es habitual que necesitemos obtener diferentes frecuencia de reloj para diferentes subsistemas. Un ejemplo muy claro puede ser el de un reloj digital que tiene que contar los segundos, por lo tanto, necesita un reloj de 1Hz (un pulso por segundo).

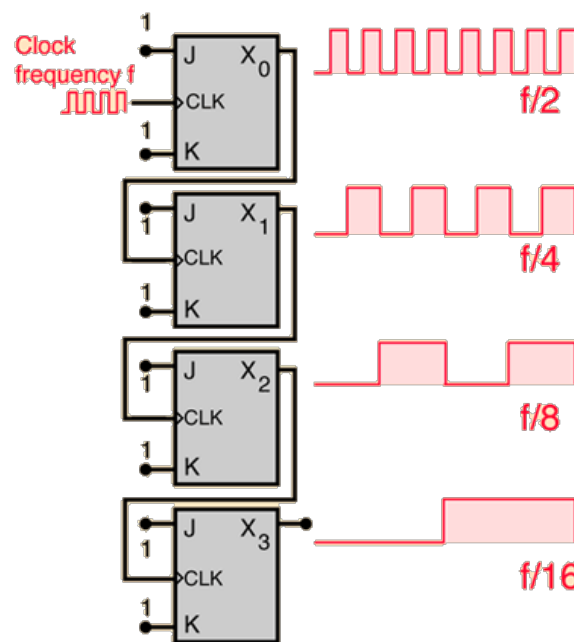


Figura 1: Frecuencias de operación de un reloj

3. Previo

4. Desarrollo

Para esta práctica tuvimos que conocer las características internas de la FPGA, por ejemplo, en la hoja de datos de la FPGA nos indica que el reloj interno opera a una frecuencia de 50 MHz, dicha frecuencia es muy alta para poder implementar un contador por lo cual tenemos que reducirla y para ello tenemos que implementar un *divisor de tiempo*.

Para implementar un divisor de tiempo debemos recordar $F = \frac{1}{T}$

$$T = \frac{1}{50MHz}$$

$$T = 20ns$$

$$\frac{1.51seg}{20 nseg} = 75500000$$

En donde 1.51 seg es el tiempo propuesto.

Ahora bien, tenemos que encontrar una forma de que el número 75 500 000 pueda ser usado como contador, para ello utilizamos el número en su forma hexadecimal 48009E0, y posteriormente podemos pasarlo a su forma binaria la cual es, 0100 1000 0000 0000 1001 1110 0000, este número se representa con 27 bits, entonces usaremos el bit 26 para representar el flanco ascendente como se muestra en el siguiente código.

```

1
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4 use IEEE.STD_LOGIC_ARITH.ALL;
5 use IEEE.STD_LOGIC_UNSIGNED.ALL;
6
7 --- Uncomment the following library declaration if instantiating
8 --- any Xilinx primitives in this code.
9 ---library UNISIM;
10 ---use UNISIM.VComponents.all;
11
12 entity divt is
13     Port ( Reloj : in  STD_LOGIC;
14           salidascount: out STD_LOGIC_VECTOR(4 downto 0);
15           CLK : out  STD_LOGIC);
16 end divt;
17
18 architecture Behavioral of divt is
19
20 begin
21 process(Reloj)
22     variable cuenta: std_logic_vector(4 downto 0):="00000";
23 BEGIN
24     sa nt <= cuenta;
25     if rising_edge(Reloj) then
26         if cuenta = "10100" then ---20 en decimal
27             cuenta:= "00000";
28         else
29             cuenta:=cuenta+1;
30         end if;
31     end if;

```

```
32     CLK<=cuenta(3);  
33  
34 end process;
```

5. Conclusiones